

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

THIS PAGE BLANK (USPTO)

①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

⑫ Pat ntschrift
⑩ DE 38 03 525 C 2

⑤① Int. Cl.⁵:
G 04 C 13/02
G 04 C 11/00
G 06 F 1/14

②① Aktenzeichen: P 38 03 525.1-31
②② Anmeldetag: 5. 2. 88
④③ Offenlegungstag: 17. 8. 89
④⑤ Veröffentlichungstag
der Patenterteilung: 2. 12. 93

Innerhalb von 3 Monaten nach Veröffentlichung der Erteilung kann Einspruch erhoben werden

⑦③ Patentinhaber:

Licentia Patent-Verwaltungs-GmbH, 60596 Frankfurt,
DE

⑦② Erfinder:

Dorn, Willi, 6277 Bad Camberg, DE; Speh, Rainer,
Dr., 6000 Frankfurt, DE

⑤⑥ Für die Beurteilung der Patentfähigkeit
in Betracht gezogene Druckschriften:

DE-OS 21 39 077
MARKUS, Norbert, RÖGNER, Rainer: Absolutzeit:
Ein Übertragungssystem der Zukunft. In: nachrichten
elektronik (1979), Nr. 7, S. 233, 234;

⑤④ Vorrichtung zum Betrieb von absoluten Echtzeituhren in einem eine Zentraluhr und Teilnehmer enthaltend n
Prozeßsteuersystem

DE 38 03 525 C 2

DE 38 03 525 C 2

Beschreibung

Die Erfindung bezieht sich auf ein Prozeßsteuersystem mit einer Reihe von Teilnehmern, die jeweils Hardware-Zeitgeber enthalten und durch Bussysteme miteinander verbunden sind, über die Informationen nach vorgegebenen Schnittstellenprozeduren übertragen werden, mit denen Interrupts in den Teilnehmern auslösbar sind.

Es ist bereits ein Absolutzeit-Übertragungssystem bekannt, bei dem von einer Hauptuhr Zeitangaben drahtgebunden an nachgeordnete Einrichtungen, wie z. B. Nebenuhren und Zeitregistriergeräte, übertragen werden (nachrichten elektronik 7-1979, S. 233, 234). Die Nebenuhren und Zeitregistriergeräte werden bei diesem Absolutzeit-Übertragungssystem unmittelbar durch Zeittakte gesteuert, die in Form eines seriellen Impulstelegramms übertragen und von den zeitanzeigenden und zeitregistrierenden Einrichtungen erkannt und in eine Zeitanzeige umgewandelt werden.

Bekannt ist auch ein Steuernetz für Zeitanzeigegeräte, bei dem zeitkennzeichnende Signale leitungsgebunden über ein beliebig organisiertes Leitungsnetz zu Anzeigegeräten übertragen werden. Die zeitkennzeichnenden Signale werden digital über eine beliebige Modulationsart einem höherfrequenten Träger aufgeprägt (DE-OS 21 39 077).

In einem Prozeßsteuersystem mit einer Reihe von Teilnehmern, die jeweils Prozeßgrößen messen, überwachen, steuern oder regeln, ist es öfters notwendig, für bestimmte Ereignisse des Prozesses die Zeit des Auftretens zu kennen bzw. bestimmte Ereignisse zu festgelegten Zeiten auszulösen. Es muß daher in Funktionseinheiten des Prozeßsteuersystems (insbesondere den prozeßnahen Einheiten) möglich sein, die Ereignisse mit einer allgemein gültigen Echtzeit in Beziehung zu bringen. Diese Teilnehmer bzw. Funktionseinheiten benötigen deshalb einen direkten Zugriff auf eine Echtzeituhr, die zugleich das Datum angibt.

Der Erfindung liegt die Aufgabe zugrunde, ein Prozeßsteuersystem der eingangs beschriebenen Gattung dahingehend weiterzuentwickeln, daß die Teilnehmer des Prozeßsteuersystems ohne Erhöhung des Leitungsaufwands für die Informationsübertragung zwischen den Teilnehmern unter weitgehender Nutzung der ohnehin für andere Zwecke erforderlichen Hardware bedarfsweise über die Echtzeit verfügen.

Die Aufgabe wird erfindungsgemäß dadurch gelöst, daß in einem Teilnehmer eines Busses für die Echtzeit eine Zentraluhr vorgesehen ist, daß in den anderen Bus-Teilnehmern gegebenenfalls Echtzeituhren vorgesehen sind, die jeweils eine von einem Prozessor geführte Software-Uhr enthalten, deren Weiterschaltung vom Prozessor durch Zugriff auf einen Hardware-Zeitgeber vorgenommen wird, und daß zur Synchronisierung der Uhren dieser Teilnehmer eine Synchronisationsprozedur vorgesehen ist, durch die jeweils die Echtzeit der Zentraluhr festgehalten und in einem definierten Zeitabstand hierzu in wenigstens einem Teilnehmer ein Interrupt ausgelöst wird, durch dessen Auftreten die im Teilnehmer vorhandene Echtzeit ebenfalls festgehalten und mit der vom Echtzeitverteiler in das Datenformat der Schnittstelle umgesetzten und zum Teilnehmer übertragenen Echtzeit der Zentraluhr zum Nachstellen der Echtzeituhr im Teilnehmer verwendet wird.

Die Zentraluhr wird z. B. mit der amtlichen Zeit eingestellt und synchronisiert. Die Zentraluhr kann durch eine Funkuhr realisiert sein. Es ist zweckmäßig, die

Echtzeit in dem oder den Teilnehmern unter Ausnutzung vorhandener Taktgeber und gegebenenfalls Zählbausteinen in Verbindung mit Software bereitzustellen. Auf diese Weise können alle Teilnehmer bzw. Funktionseinheiten, die die Echtzeit benötigen, eine Echtzeituhr führen. Beispielsweise werden alle zu einer Station einer Energieverteilungsanlage gehörenden Echtzeituhren von der zentralen Echtzeituhr gestellt und synchronisiert. Die Wirkungsrichtung entspricht hierbei vorzugsweise der Hierarchie von Datenbussystemen zwischen den Teilnehmern.

Die Echtzeit wird bereitgestellt, indem die Echtzeituhren in den Teilnehmern geführt werden, deren Echtzeit über Bussysteme verteilt wird, wobei unter Verteilung die Übertragung der Echtzeit zu den Teilnehmern und die Synchronisation der Echtzeituhren in den Teilnehmern zu verstehen ist. Die Zentraluhr ist vorzugsweise mit dem Echtzeitaufbereiter zu einer Echtzeitzentrale kombiniert, die insbesondere in einer Baueinheit realisiert ist.

Die Zentraluhr liefert die Echtzeit normalerweise in vorgegebenem Datenformat und vorzugsweise einen Sekundentakt zur Synchronisation an einen zentralen Echtzeitaufbereiter. Dieser setzt die Echtzeitcodierung in ein für die gesamte Echtzeitbereitstellung gültiges Format um. Von hier aus wird die Echtzeit über die vorhandenen Datenschnittstellen zu den einzelnen Echtzeituhren übertragen. Die Synchronisation der Echtzeituhren erfolgt zweckmäßigerweise über ohnehin vorhandene oder notwendige Hardware-Schnittstellen, z. B. Datenschnittstellen, Bussysteme und dgl., über die auch andere Informationen zwischen den Teilnehmern übertragen werden. Es ist aber auch möglich, die Synchronisation über einen separaten Sekundentakt mit zusätzlicher Hardware zu erzeugen. Dieser Sekundentakt wird dann zentral generiert. Liefert in diesem Fall nicht schon die Zentraluhr einen für das System geeigneten Sekundentakt, d. h. einen Sekundentakt mit der erforderlichen Zeitgenauigkeit, dem richtigen Pegel, der Valenz und dem fan out usw., dann wird dieser im zentralen Echtzeitaufbereiter generiert. Beide Synchronisationsverfahren können auch nebeneinander, d. h. gemischt, eingesetzt werden.

Bei der Realisierung der einzelnen Software-Funktionen müssen die Wirkungen von Unterbrechungen (Interrupts, DMA-Zugriffe) und Überlappungen (Zugriffe auf Dual-Port-Ram) besonders beachtet werden.

Mit der Erfindung wird eine Echtzeituhr zur Verfügung gestellt, die mit üblicher Weise in Funktionsbausteinen für Prozeßsteuerungen vorhandenen Hardwarekomponenten auskommt.

Jede Echtzeituhr besteht aus der Kombination eines Hardware-Zeitgebers und einer Software-Uhr. Beide zusammen bilden eine Echtzeituhr, die bei Ausfall der Synchronisierung frei weiterläuft.

In Abhängigkeit von den Hardwaregegebenheiten der Funktionsbausteine und den Softwareanforderungen können zwei verschiedene Arten von Echtzeituhren realisiert werden:

— Eine Echtzeituhr mit einem asynchronem Hardware-Zeitgeber. Diese Uhr kann normalerweise mit der vorhandenen Hardware realisiert werden und wird bevorzugt eingesetzt.

— Eine Echtzeituhr mit synchronem Hardware-Zeitgeber. Diese Uhr kann einen Zeitscheiben-Interrupt generieren. Sie sollte nur dann realisiert werden, wenn das Softwaresystem unbedingt einen

Zeitscheiben-Interrupt erfordert, da hier höhere Anforderungen an die Hardware (Zeitgeber) gestellt werden müssen.

Die Echtzeituhr mit einem asynchronen Hardware-Zeitgeber ist normalerweise auf allen Baueinheiten realisierbar, da sie keine besonderen Anforderungen an den Hardware-Zeitgeber stellt. Wenn die durch den jeweiligen Prozeß gegebenen Bedingungen es zulassen, alle Echtzeituhren mit asynchronem Hardware-Zeitgeber zu betreiben, dann reicht ein einziges Grundkonzept für sämtliche Echtzeituhren aus.

Der Hardware-Zeitgeber ist insbesondere ein binärer Zählerbaustein, auf den der Prozessor, der die Software-Uhr führt, lesend zugreifen kann. Er wird dauernd mit einem quarzstabilen Takt fast beliebiger Frequenz versorgt und zählt ohne Unterbrechung zyklisch die Taktimpulse. Die Bezeichnung "asynchron" bedeutet hier, daß die Zähler-Eingangsfrequenz und der Zählerzyklus in keinem "synchrone" Verhältnis zu irgendwelchen Zeiteinheiten der Software-Uhr stehen müssen. Diese Zählerbetriebsart ist die einfachste überhaupt und wird von allen Hardware-Zeitgebern beherrscht.

Der Hardware-Zeitgeber ist wenigstens ein aus einer Taktquelle gespeister, frei laufender Dualteiler. Beispielsweise ist der Hardware-Zeitgeber ein einfacher Dualteiler (Binärteiler), der in

- zyklischem Dauerbetrieb arbeitet und mit einer
- Eingangsfrequenz $> = 10$ MHz beaufschlagt wird, die eine hohe
- Frequenzstabilität, z. B. mit Hilfe eines Quarzes, aufweist

Die Mindest-Teilerlänge ist abhängig von der Software, da der Prozessor, der die Software-Uhr aus dem Hardware-Zeitgeber ableitet, alle Zeitgeber-Überläufe erkennen muß.

Die Stellung des Zeitgebers muß durch den Prozessor lesbar sein.

Es ist günstig, wenn ein Zeitgeber-Überlauf-Interrupt den Prozessor beaufschlagt. Durch die Software wird vorzugsweise gewährleistet, daß jeder Zeitgeber-Überlauf sicher erkannt wird. Die Nutzung eines Zeitgeber-Überlauf-Interrupts erhöht die Sicherheit und/oder die Zuverlässigkeit der Zeitmessung.

Die Zählrichtung kann beliebig sein. Bei einem Abwärtszähler muß der Zählerstand vor der Weiterverarbeitung negiert bzw. invertiert werden.

Die Software-Uhr wird in den einzelnen Funktionsbausteinen jeweils von dem Prozessor bearbeitet, der die Uhrzeit benötigt, sei es, um Ereigniszeiten zu ermitteln, Echtzeitreaktionen zu veranlassen oder auch nur um die Uhrzeitsynchronisation an nachgeordnete Uhren weiterzuleiten. Der Prozessor hat unmittelbaren Zugriff auf einen hierfür bereitgestellten Hardware-Zeitgeber.

Die Uhrzeit wird jeweils bei Bedarf und in Pausenzeiten (wenn sonst keine Prozessorleistung verlangt wird) ermittelt und vorzugsweise in einem RAM bereitgehalten (Uhrzeitführung). Die Synchronisation der Uhrzeit erfolgt zweckmäßigerweise interruptgesteuert.

Die Uhrzeitführung ist verantwortlich für den "Gang" der Echtzeituhr und stellt sicher, daß die aktuelle Echtzeit immer dann mit der geforderten Genauigkeit verfügbar ist, wenn sie benötigt wird.

Dies wird bei gegebener Hardware mit einem Minimum an Programmlaufzeit erreicht, indem folgendes

Verfahren zum Einsatz kommt: Der Prozessor, der die Software-Uhr führt, kann den Hardware-Zeitgeber direkt lesen.

Zur Uhrzeitführung wird zunächst der Hardware-Zeitgeber eingelesen.

Dann wird ermittelt, wieviel Zeit seit der letzten Aktualisierung der Software-Uhr vergangen ist. Hierzu wird der Stand des Hardware-Zeitgebers, der zur letzten aktuellen Software-Uhrzeit gehört, bereitgehalten.

Eine Berechnung gibt die Anzahl der Millisekunden an, die der Uhrzeit aktuell hinzugerechnet werden müssen. Wird eine fortgesetzte Subtraktion (Multiplikation) verwendet, dann ist die Programmlaufzeit zur Berechnung der neuen Uhrzeit umso größer, je mehr Zeit seit der letzten Zeitrechnung vergangen ist.

Aus diesem Grund kann es vorteilhaft sein, die Uhrzeit nicht nur bei Bedarf neu zu berechnen, sondern zwischendurch immer dann, wenn das Programmsystem eine Pausenschleife durchläuft und ohnehin keine Programmlaufzeit beansprucht.

Dieses Verfahren kann auch dazu dienen, die Überläufe des Hardware-Zeitgebers zu erfassen und entsprechend zu berücksichtigen. Diese Art der Zeitgeber-Überlauferkennung funktioniert dann, wenn sichergestellt wird, daß zwischen je zwei benachbarten Neuberechnungen der Uhrzeit weniger als eine Zeitgeber-Zykluszeit liegt.

Ein sicheres Verfahren zur Erkennung der Zeitgeber-Überläufe liefert der Zeitgeber-Überlauf-Interrupt. In der Interrupt-Routine kann entweder ganz normal die Uhrzeit berechnet werden oder einfach ein Überlaufzähler nachgezählt werden. Dieser Zähler wird dann bei einer später erfolgenden Uhrzeitberechnung berücksichtigt.

Der Rest der Uhrzeitführung wird durch das gewünschte Format der Uhrzeit wesentlich mitbestimmt und besteht im allgemeinen aus einem mehrstufigen Zähler-Aufbau mit Sonderfunktionen.

Es ist auch möglich, eine Echtzeituhr mit synchronem Hardware-Zeitgeber aufzubauen.

Bei zu steuernden Prozessen in der Energieverteilung ist es erforderlich, daß in den jeweils mit einer Echtzeituhr ausgestatteten Funktionsbausteinen eine bestimmte Zeitdifferenz zwischen zwei Uhren nicht überschritten wird, die z. B. im Bereich von 10 msec liegt.

Die Erfüllung dieser Forderung wird bei der Verteilung der Echtzeit von der Zentraluhr bis zur Echtzeituhr im "letzten" Funktionsbaustein berücksichtigt.

Die Verteilung der Echtzeit wird vorzugsweise in die beiden Verfahrensschritte "Echtzeitübertragung" und "Synchronisation der Echtzeituhren" zerlegt.

Übertragung der Echtzeit

Die Übertragung der Echtzeit erfolgt über die vorhandenen Daten-Schnittstellen, z. B. über Bussysteme von Uhr zu Uhr.

Das Datenformat bei der Übertragung der Echtzeit ist an allen Schnittstellen gleich und stellt eine Erweiterung der Echtzeitdarstellung im Prozeß dar. Es ist zugleich eine Teilmenge der Darstellung einer kompletten Software-Uhr. Das Format ist unabhängig von der Art der Synchronisation der Echtzeit-Uhren.

Eine Ausnahme bei der Echtzeitübertragung gibt es beim Datenformat der Zentraluhr. Dieses Format wird von der verwendeten Zentraluhr vorgegeben und im zentralen Echtzeitaufbereiter in das Standardformat umgesetzt, das bei allen Übertragungsschnittstellen

gleich ist.

Die Übertragung zwischen Software-Uhr und Übertragungs-Schnittstelle erfolgt blockweise, d. h. die gesamte Uhrzeit wird an einem Stück übertragen und wird gegen gleichzeitige Uhrverstellung verriegelt.

Synchronisation der Echtzeituhren

Von zentraler Bedeutung für die gesamte Echtzeitbereitstellung ist die Art der Synchronisation der Echtzeituhren. Hier kommen vorzugsweise zwei Verfahren in Betracht, die oben bereits erwähnt wurden:

- Synchronisation über die vorhandenen Schnittstellen (Bussysteme),
- Synchronisation über einen eigenen Takt (separate Taktleitung).

Das erste Verfahren erfordert mehr Softwareaufwand, während das zweite Verfahren mehr Hardware benötigt. Beide Verfahren können auch "gemischt" in einer Anlage vorkommen.

Das erste Verfahren ist wirtschaftlicher.

Beide Verfahren sind so aufgebaut, daß sie Ausfälle von "Synchronimpulsen" tolerieren.

Synchronisation über die Datenschnittstellen

Bei diesem Verfahren wird die Synchronisation über eine Schnittstellenaktion zwischen Teilnehmern einer Schnittstelle erreicht. Schnittstellen sind hier die Verbindungen, über die auch die Grobuhrzeit übertragen wird. Es wird dabei keine Zusatzhardware benötigt.

Für jede Schnittstelle wird eine "Synchronisationsprozedur" definiert. Es ist möglich, für alle Bus-Schnittstellen (parallel und seriell) ein prinzipielles Konzept zur Uhrzeitsynchronisation anzugeben:

Wichtigstes Prinzip bei der Synchronisation über die Datenschnittstellen ist die Reduzierung von undefinierten Verzögerungszeiten auf das erreichbare Minimum. Dabei muß bedacht werden, daß bei einer Synchronisation über mehrere Stufen die Fehler sich addieren. Definierte Verzögerungszeiten sind keine Fehlerquellen, da sie berücksichtigt werden können.

Das Ziel, mit möglichst geringen undefinierten Verzögerungszeiten (bei der Uhrzeitsynchronisation über die Datenschnittstellen) zu arbeiten, wird wie folgt erreicht:

Alle verwendeten Schnittstellen bieten die Möglichkeit, durch bestimmte Schnittstellenereignisse (z. B. "Datenübertragung beendet") einen Interrupt zur Verarbeitungseinheit, die die Uhrzeit führt, auszulösen. Dies gilt auch für die Bus-Controller.

Es werden Schnittstellenprozeduren angewendet, die bei verschiedenen Busteilnehmern (Master, Slave) einen Interrupt auslösen. Es hat sich gezeigt, daß der zeitliche Abstand dieser Interrupte zu einem bestimmten Ereignis an der Schnittstellen-Hardware auf einige Mikrosekunden genau bestimmbar ist. Dadurch wird der zeitliche Abstand von Interrupts, die durch eine Schnittstellenprozedur bei unterschiedlichen Schnittstellenteilnehmern ausgelöst werden (z. B. Masters und Slaves), ebenfalls auf Mikrosekunden genau bestimmt.

Die Hardware-Voraussetzungen für eine ausreichend genaue Synchronisierung durch Interrupt sind durch die oben erwähnte Feststellung gegeben. Die Software wird so ausgebildet, daß der anstehende relevante Interrupt wenigstens einmal innerhalb eines bestimmten Zeitintervalls, spätestens nach einer tolerierbaren

Sperrzeit freigegeben wird.

Das Verfahren beinhaltet folgende Schritte:

1. Es wird eine spezielle Bus-Prozedur (Übertragung von einem Master bzw. einem übergeordneten Busteilnehmer zu einem Slave bzw. einer untergeordneten Einheit) mit einer (zu vereinbarenden) Kennung SYNC gestartet. Diese Prozedur kann auch ein "Aufruf an alle" sein (soweit vorhanden).
2. Die SYNC-Busprozedur löst bei allen beteiligten Busteilnehmern, auch beim Master, einen Interrupt aus.
3. Die Interrupt-Routinen frieren die augenblickliche Feinzeit in den beteiligten Teilnehmern ein.
4. Ist ein SYNC-Interrupt in einem Teilnehmer gesperrt, wenn das SYNC-Ereignis eintritt, dann wird dort die Uhrzeit erst nach der Interrupt-Freigabe eingefroren und ist möglicherweise nicht mehr aktuell (spätere Zeit). Damit dies erkennbar ist, wird auch immer dann, wenn der Interrupt gesperrt wird, die aktuelle Uhrzeit eingefroren. Ist der Interrupt nicht gesperrt, wenn er ausgelöst wird, dann wird dies erkannt, und beide Zeiten werden gleichgesetzt, womit sie als eindeutig gültig erklärt sind.
5. In einer später folgenden Busprozedur sendet der übergeordnete Busteilnehmer dem bzw. den untergeordneten Busteilnehmern seine eingefrorenen Feinzeiten zusammen mit der Grobzeit. Damit ist zugleich die Funktion "Übertragung der Echtzeit" erfüllt.
6. Die untergeordneten Busteilnehmer berechnen jeweils aus der übertragenen Uhrzeit und ihrer eigenen (eingefrorenen) Zeit die Differenz (= Regelabweichung für die eigene Uhr) und regeln ihre Uhren ein oder stellen sie neu, wenn die Regelabweichung mehrmals hintereinander unzulässig groß (asynchron) war.

Diese Prozedur muß nicht sofort durchgeführt werden und kann in Abhängigkeit von der Programmlaufzeit optimal aufgerufen werden.

Erfolgt die Synchronisation der Echtzeituhren über einen separaten Takt, dann wird dieser Takt den Prozessoren, die ihre Echtzeituhr damit synchronisieren sollen, als Interrupt-Auslösesignal zugeführt.

Die Taktfrequenz beträgt vorzugsweise genau ein Hertz (Sekundentakt). Der Beginn jeder neuen Sekunde wird durch die aktive Taktflanke angezeigt (0-1-Übergang).

Es ist von Vorteil, wenn bei beiden in Frage kommenden Synchronisationsverfahren möglichst mit denselben Softwareroutinen gearbeitet werden kann, deshalb wird eine Prozedur angewendet, die weitgehende Übereinstimmung mit dem oben angegebenen Synchronisationsverfahren aufweist. Der wesentliche Unterschied besteht darin, daß anstelle einer ereignisgesteuerten Routine eine zeitabhängig gesteuerte Routine (Interrupt-Sekundentaktroutine) durch den Sekundentakt aufgerufen wird. In dieser Routine wird für die Feinzeit (die über den Bus empfangen wird) ein sekundentaktgerechter Ersatz generiert.

In der Initialisierungsphase werden zunächst alle Variablen mit definierten Werten vorbesetzt. Das System der Echtzeitverteilung sorgt dann dafür, daß nach einer Hochlaufzeit alle Uhren synchronisiert sind und mit der geforderten Genauigkeit "gehen". Alle Zustände werden über Flags angezeigt.

Die Erfindung im folgenden anhand von in einer

Zeichnung dargestellten Ausführungsbeispielen näher beschrieben, aus denen sich weitere Einzelheiten, Merkmale und Vorteile ergeben. Es zeigt

Fig. 1 ein Schaltbild einer Anordnung mit einer Zentralen, die eine zentrale Echtzeituhr enthält, und mit der Zentralen verbundene Teilnehmer, die jeweils Echtzeituhren enthalten,

Fig. 2 ein Ablaufdiagramm für ein Hauptprogramm in einem Teilnehmer, der eine Software-Echtzeituhr und einen Hardware-Timer enthält,

Fig. 3 ein Ablaufdiagramm der Aktualisierung der Uhrzeit in einem Teilnehmer,

Fig. 4 ein Ablaufdiagramm der Feststellung des Zählstandes des Hardware-Timers in einem Teilnehmer,

Fig. 5 ein Ablaufdiagramm der Eingabe eines neuen Zählstandes in den Hardware-Timer,

Fig. 6 ein Ablaufdiagramm des Takts der Software-Echtzeituhr,

Fig. 7 ein Ablaufdiagramm für die Generierung von Millisekundentakten,

Fig. 8 ein Ablaufdiagramm für die Bearbeitung des Millisekundentakts,

Fig. 9 ein Ablaufdiagramm eines einen externen Sekundentakt zugeordneten Interrupts,

Fig. 10 ein Ablaufdiagramm eines einen Empfangstakt zugeordneten Interrupts,

Fig. 11 ein Ablaufdiagramm eines einem Uhrzeitsynchronkontakt zugeordneten Interrupts,

Fig. 12 ein Ablaufdiagramm eines einem Primäruhr-Synchronkontakt zugeordneten Interrupts.

In einer Station 4 bzw. Prozeßleitsystem ist als Zentralteil eine Stationseinheit 1 vorgesehen. Diese enthält als Steuereinheit ein Mehrrechner-System, z. B. mit parallelem Bus.

Teilnehmer 2, 3 am Systembus 11 sind Baugruppen, die je eine multimasterfähige Rechnerkarte darstellen. Jede Rechnerkarte führt eine Echtzeituhr 5, 6.

Eine Rechnerkarte der Stationseinheit 1 enthält zusätzlich die Software eines zentralen Echtzeitverteilers 12. Diese Baugruppe wird von einer Zentraluhr 7 der Station 4 aus mit der Echtzeit versorgt. Von hier aus wird die Echtzeit zu den Uhren der einzelnen Rechnerkarten über den Systembus 11 "verteilt".

Die Weiterverteilung der Echtzeit geschieht von der Stationseinheit 1 aus über die seriellen Bus-Systeme, von denen ein Bus 8 dargestellt ist. Dabei wird ein Anlagenbus normalerweise durch eine serielle Schnittstelle einer Baugruppe realisiert. Falls die notwendigen Schnittstellen auf der Baugruppe nicht zur Verfügung stehen und durch Zusatzmodule realisiert werden, die normalerweise ebenfalls einen eigenen Mikroprozessor beinhalten, müssen auf diesen ebenfalls eigene Echtzeituhren geführt werden.

Als Hardware-Timer 9, 10 wird jeweils der Timer eines Prozessors verwendet. Die Timer können sowohl prozessorintern als auch extern Timerereignisse (z. B. Timer-Überlauf) als Interrupt zum Prozessor melden.

Die Verteilung von Echtzeit über den Systembus 11 erfolgt vom zentralen Echtzeitverteiler 12 aus. Dieser ist auf einer Rechnerkarte realisiert. Von hier aus wird die Uhrzeit über den Systembus 11 zu allen anderen Teilnehmern 2, 3 übertragen und synchronisiert.

Alle beteiligten Busteilnehmer 2, 3 führen ihre eigene Echtzeituhr.

Zur Durchführung der Übertragung und Synchronisation der Echtzeit über den Systembus 11 bewirbt sich der zentrale Echtzeitverteiler 12 um die Bushoheit. Er sendet zunächst eine Meldung zur Uhrzeitsynchronisa-

tion an die Busteilnehmer 2, 3, deren Echtzeituhr zu synchronisieren ist. Diese Meldung kann alle Busteilnehmer gleichzeitig synchronisieren. In folgenden Meldungen wird dann die "eingefrorene" Uhrzeit des Echtzeitverteilers 12 zu den anderen Bus-Teilnehmern übertragen.

Der zentrale Echtzeitverteiler 12 ist auf einer Baugruppe realisiert. Von hier aus werden alle anderen Uhren in der Stationseinheit 1 über den Systembus 11 gestellt und synchronisiert.

Der zentrale Echtzeitverteiler führt hierzu eine Echtzeituhr wie alle anderen Bus-Teilnehmer.

Im Unterschied zu den anderen Uhren wird diese aber nicht von einem übergeordneten Echtzeitverteiler und synchronisiert. Das genaue Verfahren der Uhrzeiteinstellung und -synchronisation ist abhängig von der verwendeten Zentraluhr.

Zentraluhren

In jeder Stationseinheit 1 gibt es einen zentralen Echtzeitverteiler 12, der eine eigene Echtzeituhr führt und die Echtzeit über den Systembus 11 an die anderen Echtzeituhren der Stationseinheit verteilt.

Der zentrale Echtzeitverteiler selbst erhält Uhrzeit und "Synchronisierimpulse" von der Zentraluhr 7.

Die Zentraluhr 7 ist normalerweise Teil der Stationseinheit 1 und kann verschieden aufgebaut sein.

Nachfolgend werden die verschiedenen Verfahren einzeln erläutert.

Vorzugsweise empfängt eine Funkuhr den Zeit- und Normalfrequenzsender DCF77 und stellt Uhrzeit und Datum über verschiedene Schnittstellen zur Verfügung. Werden die Schnittstellen durch die Funkuhr aktiviert, dann werden die Schnittstellen-Ereignisse uhrzeitsynchron ausgelöst. Die absolute Wiederholgenauigkeit beträgt hierbei 20 Millisekunden (abhängig von der Empfangsfeldstärke) bei einem Kurzzeitjitter von einer Millisekunde.

Die Funkuhr 7 sendet selbständig alle Sekunden Uhrzeit und Datum über ihre serielle Schnittstelle zum Echtzeitverteiler 12.

Für den zentralen Echtzeitverteiler 12 bedeutet dies, daß alle Sekunden durch den Sekundentakt einen Interrupt ausgelöst wird. Darauf folgt die komplette Übertragung der Uhrzeit.

Der jeweilige Systembus-Teilnehmer 2, 3 und eventuell vorhandene Zusatzmodul n sind über eine gemeinsame Schnittstelle miteinander gekoppelt. Alle Module enthalten je eine eigene Echtzeituhr. Dabei wird die Uhr auf dem Zusatzmodul von der (übergeordneten) Uhr über die Schnittstelle gestellt und synchronisiert. Auch hier wird grundsätzlich das unten noch näher erläuterte Übertragungsverfahren angewendet.

Lediglich die "Synchronisation über die Datenschnittstelle" wird abgeändert, da es kein Hardware-Schnittstelleneignis gibt, das auf beiden Seiten der Schnittstelle einen Interrupt auslöst. Hier löst lediglich ein Teilnehmer 2 bzw. 3 auf einer Seite der Schnittstelle einen Interrupt beim Teilnehmer auf der anderen Seite der Schnittstelle aus. Als Ersatz für den Interrupt der auftretenden Seite wird dort unmittelbar nach dem Auslösen (des Interrupts zur anderen Seite) eine Interrupt-Ersatz-Routine aufgerufen, also so getan, als wäre ein Interrupt der Schnittstellenprozedur ausgelöst worden. Die Richtung, in der dieser Vorgang über die Schnittstelle abläuft, ist prinzipiell beliebig, sollte jedoch möglichst der Systemhierarchie entsprechen.

Die Verteilung der Echtzeit über den Anlagenbus 8 erfolgt vom Teilnehmer 2 oder 3 aus.

Alle beteiligten Busteilnehmer (Master und Slaves) führen ihre eigene Echtzeituhr.

Zur Durchführung der Uhrzeitübertragung und -synchronisation sendet der Master zunächst ein Telegramm, das bei den aktiven Bus-Teilnehmern einen Interrupt auslöst.

Später wird vom Master die "eingefrorene" Zeit zu dem Slave übertragen.

Die genaue Zeitverschiebung zwischen dem Interrupt des Masters und dem Interrupt des Slaves wird einmalig experimentell ermittelt.

Die Echtzeit wird der Feldeinheit über den Anlagenbus 8 zugeführt. Auch die Synchronisation der Uhrzeit in der Feldeinheit erfolgt über den Anlagenbus.

In den Teilnehmer 2, 3 läuft jeweils ein Hauptprogramm ab, das eine Software-Uhr enthält. In Fig. 2 ist ein Ablaufdiagramm in Verbindung mit der Software-Uhr dargestellt.

Nach dem Start werden die zu steuernden Uhren in einem Schritt 13 initialisiert. Danach wird in eine Hauptprogrammschleife 14 eingetreten, die folgende Programmteile enthält: Echtzeit von extern übernehmen, Ausgabe der Uhrzeit, SYNC-Interrupt Disable bzw. SYNC-INTERRUPT ENABLE, Disable bzw. enable speziellen Interrupt und disable bzw. enable generellen Interrupt. Wenn nach dem Durchlaufen der Hauptprogrammschleife noch Programmzeit vorhanden ist, wird in einem Programmteil 15 die Echtzeituhr korrigiert. Ansonsten wird die Hauptprogrammschleife wiederholt. Verbleibt nach der Korrektur der Uhr noch Zeit, wird die Echtzeituhr in einem Programmteil 16, das in Fig. 3 dargestellt ist.

Das Programmteil 16 enthält die Programmteile 17, 18, 19, 20 jeweils für Interrupt disable, Hardware-Timer lesen, Uhrzeit neu berechnen und Interrupt enable.

Die Uhrzeitführung durch Hardware-Timer lesen, geht aus Fig. 4 hervor. Dieses Unterprogramm enthält die Teile Interrupt disable, Timer Input 21, die Abfrage nach dem Überlauf des Timers nach dem letzten Einlesen, die Korrektur 22 des Zählerstands bei einem Überlauf und dem Schnitt Interrupt enable. Der Aufbau des Unter-Programms 21 ist in Fig. 5 dargestellt. Das Unterprogramm beinhaltet die Erstellung einer Kopie des Hardware-Timers und, wenn die einzelnen Zählerwerte synchron gelesen werden müssen, die temporäre Speicherung der Wörter. Es wird der Stand des Hardware-Timers, der der letzten aktuellen Software-Uhrzeit entspricht, bereitgehalten. Die Aktualisierung des Software-Uhr-Takts ist in Fig. 6 dargestellt. Diese Aktualisierung beinhaltet folgende Unterprogramme: Millisekundengenerierung 23, Millisekundentaktbearbeitung 24, Minutentaktbearbeitung 25, Stundentaktbearbeitung 26, Tagestaktbearbeitung 27, Monatstaktbearbeitung 28, Jahrestaktbearbeitung 29 und Uhrzustandtest 30.

Das Unterprogramm 23 setzt zunächst den Millisekundentakt auf null und prüft danach, ob der Hardware-Timer um 1 ms weitergezählt hat. Wenn dies der Fall ist, wird die Hardware-Timer-Nachführung um 1 ms erhöht und anschließend 1 ms für die Software-Uhr registriert, bevor die Prüfung wiederholt wird. Hat der Hardware-Timer nicht um 1 ms weitergezählt, dann wird der Timer-Rest mit dem Stand der Software-Uhr verrechnet.

Die Verarbeitung der Millisekundentakte ist in Fig. 8 dargestellt. Nach dem in Fig. 8 dargestellten Verfahren werden Minutentakte aus Millisekundentakte erzeugt. Die Verarbeitung der Minutentakte, Stundentakte, Ta-

gestakte, Monatstakte und Jahrestakte erfolgt entsprechend und ist nicht näher dargestellt.

Im System wird durch die getrennte Übertragung der Echtzeit zu den Teilnehmern und der Synchronisationsinformation eine Verminderung der Belegung der Übertragungskanäle durch die Echtzeitverteilung erreicht. Die Echtzeit wird an allen Schnittstellen mit dem gleichen Datenformat übertragen. Das Datenformat ist daher unabhängig von der Art der Synchronisation der Echtzeituhr. Die Echtzeit wird blockweise über die jeweilige Schnittstelle und in einem Stück übertragen, wobei die gleichzeitige Uhrverstellung verriegelt wird. Die Echtzeitkorrektur im jeweiligen Teilnehmer beginnt mit einem Aufruf durch die Uhrzeit-Übergabe. Danach wird die berechnete Uhrzeit mit der Sollzeit verglichen. Die Uhrzeit ist unterteilt in eine Grobzeit und eine Feinzeit. Es wird geprüft, ob die berechnete Uhrzeit innerhalb zulässiger Toleranzen mit der übertragenen Uhrzeit übereinstimmt. Falls dies nicht zutrifft, wird die Zeit in der Echtzeituhr des jeweiligen Teilnehmers korrigiert.

Vorzugsweise werden die Echtzeituhrn über die vorhandenen Schnittstellen synchronisiert. Für die Schnittstellen wird eine Synchronisationsprozedur definiert. Alle verwendeten Schnittstellen bieten die Möglichkeit, durch bestimmte Schnittstelleneignisse (insbesondere "Übertragung beendet") einen Interrupt zu dem Prozessor, der die Uhrzeit führt, auszulösen.

Für die Synchronisierung wird eine spezielle Bus-Prozedur (Übertragung vom Master zum Slave) mit einer vorgebbaren Kennung SYNC gestartet. Diese Prozedur kann ein Aufruf an alle sein. Die SYNC-Prozedur löst bei allen beteiligten Busteilnehmern (auch beim Master) einen Interrupt aus. Für die Interrupt-Auslösung kann ein externer Sekundentakt, ein Empfangstakt, ein Uhrzeitsynchronkontakt oder ein Primäruhr-Synchronkontakt vorgesehen sein. Je nach dem für die Interruptauslösung bestimmten Takt ergeben sich verschiedene Verfahrensschritte. Nach der in Fig. 9 gezeigten, durch einen externen Sekundentakt bewirkten Synchronisierung schließt sich an den Interrupt-Schritt ein Unterprogramm 31 "Uhrzeit bearbeiten" an, nach dessen Ende in einem nachfolgenden Schritt 32 ein Flag gesetzt wird, das einen Synchronkontakt als "Sekundentakt" kennzeichnet. Danach wird die Soll-Feinzeit anhand des Sekundentakts berechnet. Anschließend wird die Differenz zwischen dem Zählerstand und dem Rest null gesetzt, woraus sich eine neue Feinzeit ergibt.

Bei einem Interrupt mit Hilfe eines Empfangstakts wird das in Fig. 10 dargestellte Verfahren durchlaufen. Es folgt auf den Interrupt Schritt wiederum das Unterprogramm 31, dem sich ein Schritt 35 anschließt, in dem ein Flag den Synchronkontakt als Empfangstakt kennzeichnet. Das in Fig. 11 dargestellte Unterprogramm 31 enthält einen Interrupt-Schritt, der unmittelbar durch einen Uhrzeitsynchronkontakt ausgelöst werden kann. An diesen Schritt schließt sich das Unterprogramm 16, Uhrzeitaktualisieren an, auf das ein Unterprogramm 35 "Feinzeit merken" folgt. Danach wird in einem Schritt 36 der Synchronkontakt gezählt. In einem Schritt 37 wird ein Flag gesetzt, wenn die Kennung SYNC während einer Interruptsperrung erzeugt wurde.

Wenn ein Interrupt durch einen Primäruhr-Synchronkontakt erzeugt wird, wie er z. B. von einem Empfänger eines Zeitsenders erzeugt wird, dann wird auf den Interrupt-Schritt wiederum die Uhrzeit durch das Unterprogramm 16 aktualisiert. Danach folgt das Unterprogramm 36, nach dessen Ende ein Interrupt-Request

durchgeführt wird, der sich auf eine Interruptsperrung bezieht. Wenn keine Interruptsperrung beim Eintreten des SYNC-Ereignisses eintritt, wird der Interrupt zu einer Zieladresse zugeordnet, wonach die Feinzeit und die Grobzeit für die Ausgabe zur Verfügung stehen. Bei einer Interruptsperrung wird die Uhrzeit erst nach der Interrupt-Freigabe eingefroren, bevor die weiteren oben erwähnten Schritte folgen.

In einer anschließenden Busprozedur sendet der Master dem oder den Slaves seine eingefrorenen Feinzeiten zusammen mit der Grobzeit. Damit ist die Übertragung der Echtzeit beendet.

Jeder Slave wird vorzugsweise mindestens mehrmals pro Minute synchronisiert. Die Slaves berechnen aus der übertragenen Uhrzeit und ihrer eigenen eingefrorenen Zeit die Zeitdifferenz für die eigene Echtuhr und regeln ihre Uhren nach bzw. stellen sie neu ein. Wenn die Regelabweichung mehrmals nacheinander unzulässig groß war, wird eine Testprozedur zur Fehlersuche angeführt. Die Testprozedur muß nicht sofort durchgeführt werden, sondern kann unter Berücksichtigung der Programmlaufzeit optimal aufgerufen werden.

In der Initialisierungsphase werden alle Variablen mit definierten Werten besetzt. Die Echtzeitverteilung sorgt dafür, daß nach einer Hochlaufzeit alle Echtzeithren synchron laufen. Alle Zustände werden durch Flags angezeigt.

Patentansprüche

1. Prozeßsteuersystem mit einer Reihe von Teilnehmern, die jeweils Hardware-Zeitgeber enthalten und durch Bussysteme miteinander verbunden sind, über die Informationen nach vorgegebenen Schnittstellenprozeduren übertragen werden, mit denen Interrupts in den Teilnehmern auslösbar sind, **dadurch gekennzeichnet**, daß in einem Teilnehmer eines Busses (1, 2, 3) für die Echtzeit eine Zentraluhr (7) vorgesehen ist, daß in den anderen Busteilnehmern gegebenenfalls Echtzeithren (5, 6) vorgesehen sind, die jeweils eine von einem Prozessor geführte Software-Uhr enthalten, deren Weiterschaltung vom Prozessor durch Zugriff auf einen Hardware-Zeitgeber vorgenommen wird, und daß zur Synchronisierung der Uhren dieser Teilnehmer (2, 3) eine Synchronisationsprozedur vorgesehen ist, durch die jeweils die Echtzeit der Zentraluhr (7) festgehalten und in einem definierten Zeitabstand hierzu in wenigstens einem Teilnehmer (2, 3) ein Interrupt ausgelöst wird, durch dessen Auftreten die im Teilnehmer vorhandene Echtzeit ebenfalls festgehalten und mit der vom Echtzeitverteiler in das Datenformat der Schnittstelle umgesetzten und zum Teilnehmer übertragenen Echtzeit der Zentraluhr zum Nachstellen der Echtzeithren im Teilnehmer (2, 3) verwendet wird.

2. Prozeßsteuersystem nach Anspruch 1, dadurch gekennzeichnet, daß die Hardware-Zeitgeber in den Teilnehmern (2, 3) frei laufend und somit auch für andere Zwecke nutzbar oder programmierbar sind und daß jeweils aus der Zählstandsdifferenz des Hardware-Zeitgebers das Weiterschalten der Software-Uhr bestimmt wird.

3. Prozeßsteuersystem nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß der Hardware-Zeitgeber mindestens ein aus einer Taktquelle gespeister, frei laufender Dualteiler ist.

4. Prozeßsteuersystem nach einem der vorherge-

henden Ansprüche, dadurch gekennzeichnet, daß die Taktquelle nur eine geringe Langzeitstabilität haben muß.

5. Prozeßsteuersystem nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß der Überlauf des jeweiligen Hardware-Zeitgebers (9, 10) im Teilnehmer (2, 3) die Berechnung der Echtzeit auslöst oder einen Software-Überlaufzähler beaufschlagt, dessen Inhalt bei später durchgeführter Berechnung berücksichtigt wird, oder daß der Hardware-Zeitgeber durch den Prozessor in so kurzen Zeitabständen gelesen wird, daß der Überlauf aufgrund der Zeitabstände sicher erkannt wird.

6. Prozeßsteuersystem nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß in jedem übergeordneten Busteilnehmer durch ein n Interrupt das Festhalten der aktuellen Uhrzeit synchronisiert und die festgehaltene Uhrzeit zu untergeordneten Einheiten übertragen wird und in diesen jeweils mit einer dort durch einen entsprechenden Interrupt desselben Busereignisses festgehaltenen Uhrzeit verglichen wird und daß die Differenz zur Korrektur der Ganggeschwindigkeit der Echtzeithren in der untergeordneten Einheit verwendet wird.

Hierzu 10 Seite(n) Zeichnungen

- Leerseite -

THIS PAGE BLANK (USPTO)

FIG.1

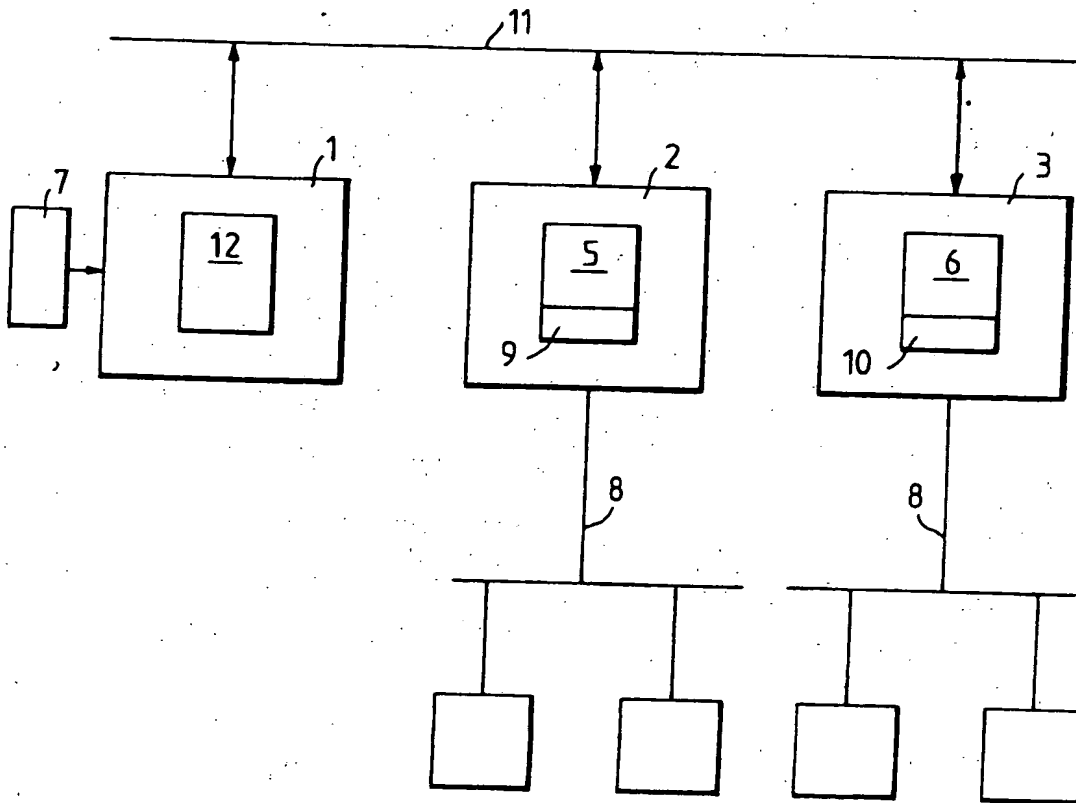
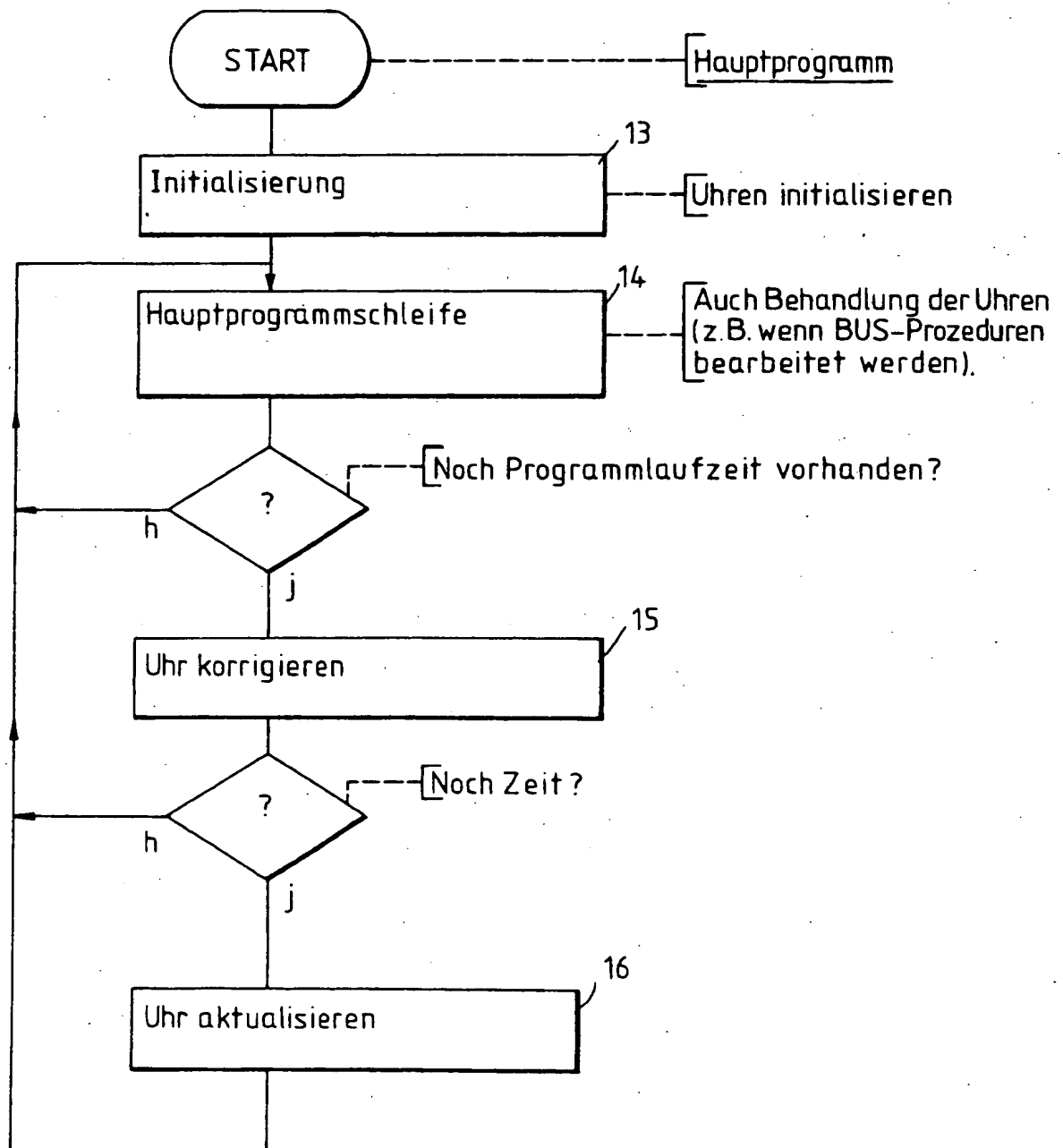


FIG.2



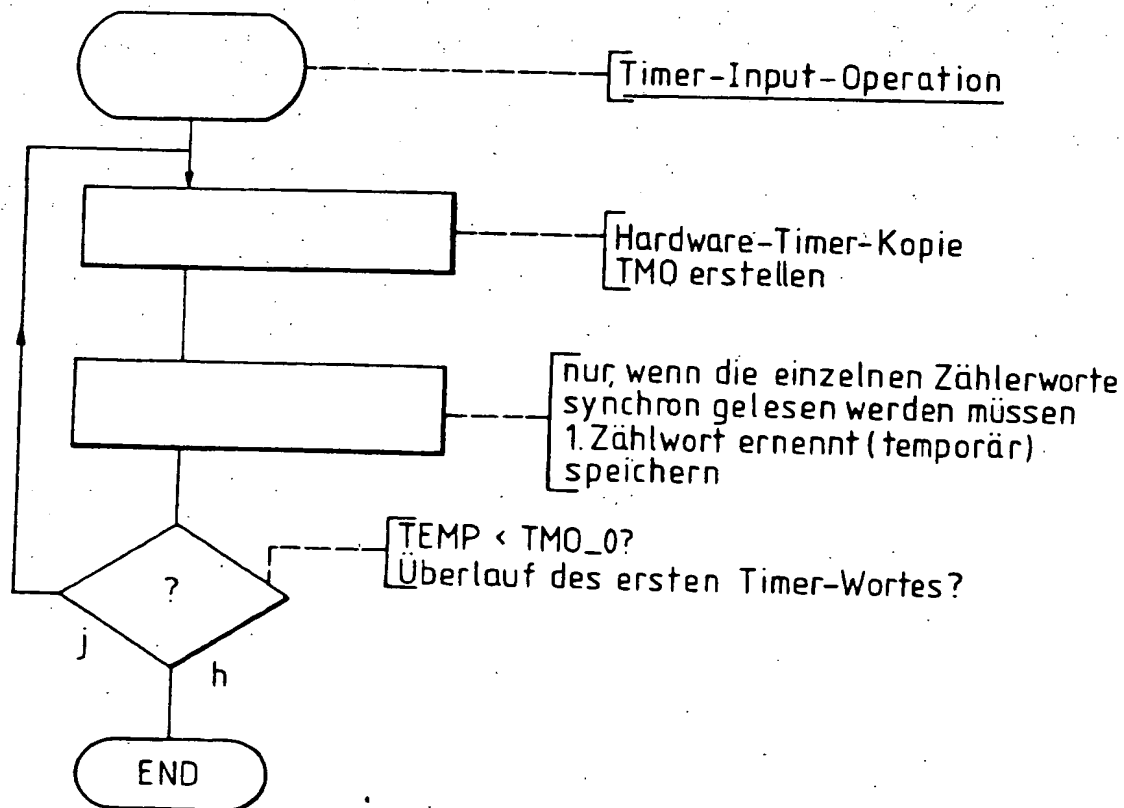
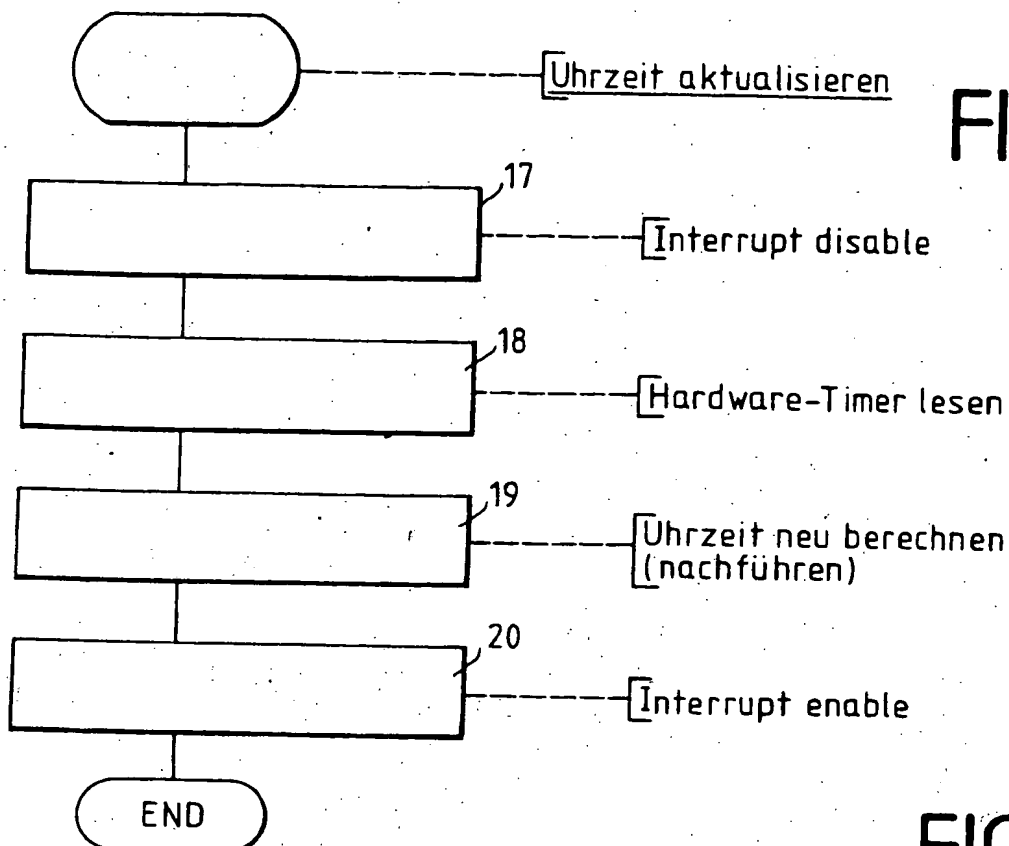
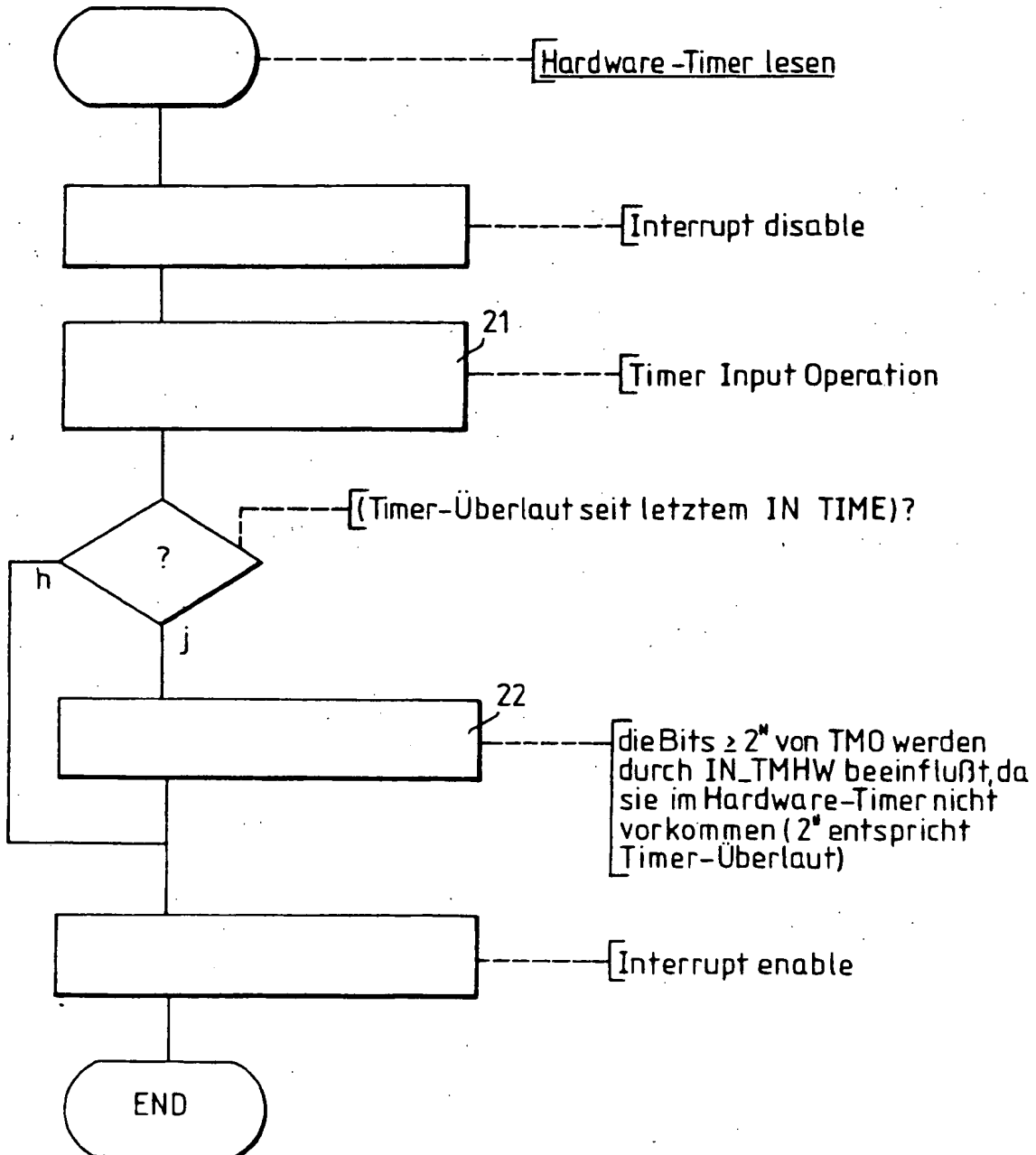


FIG.4



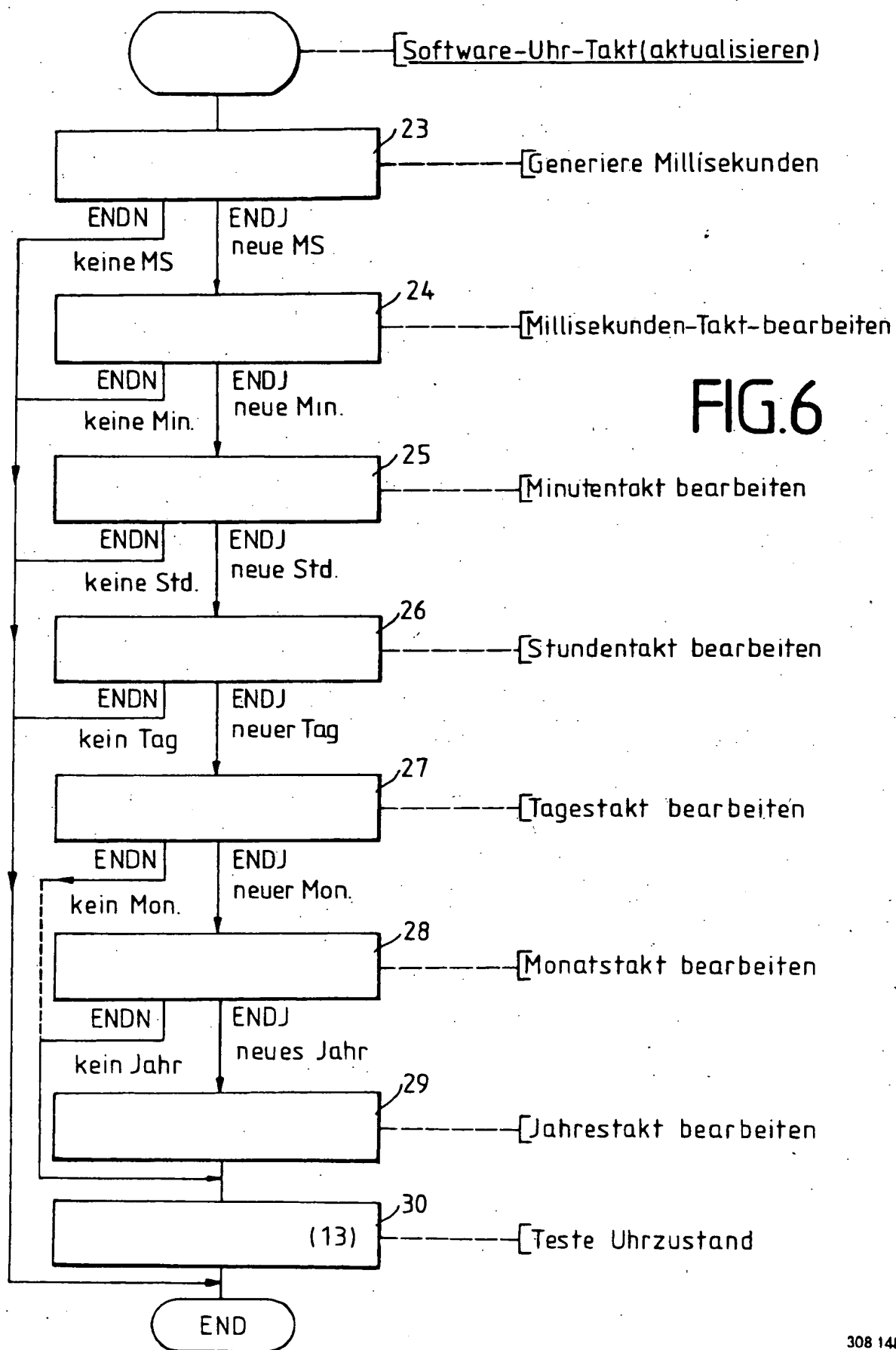
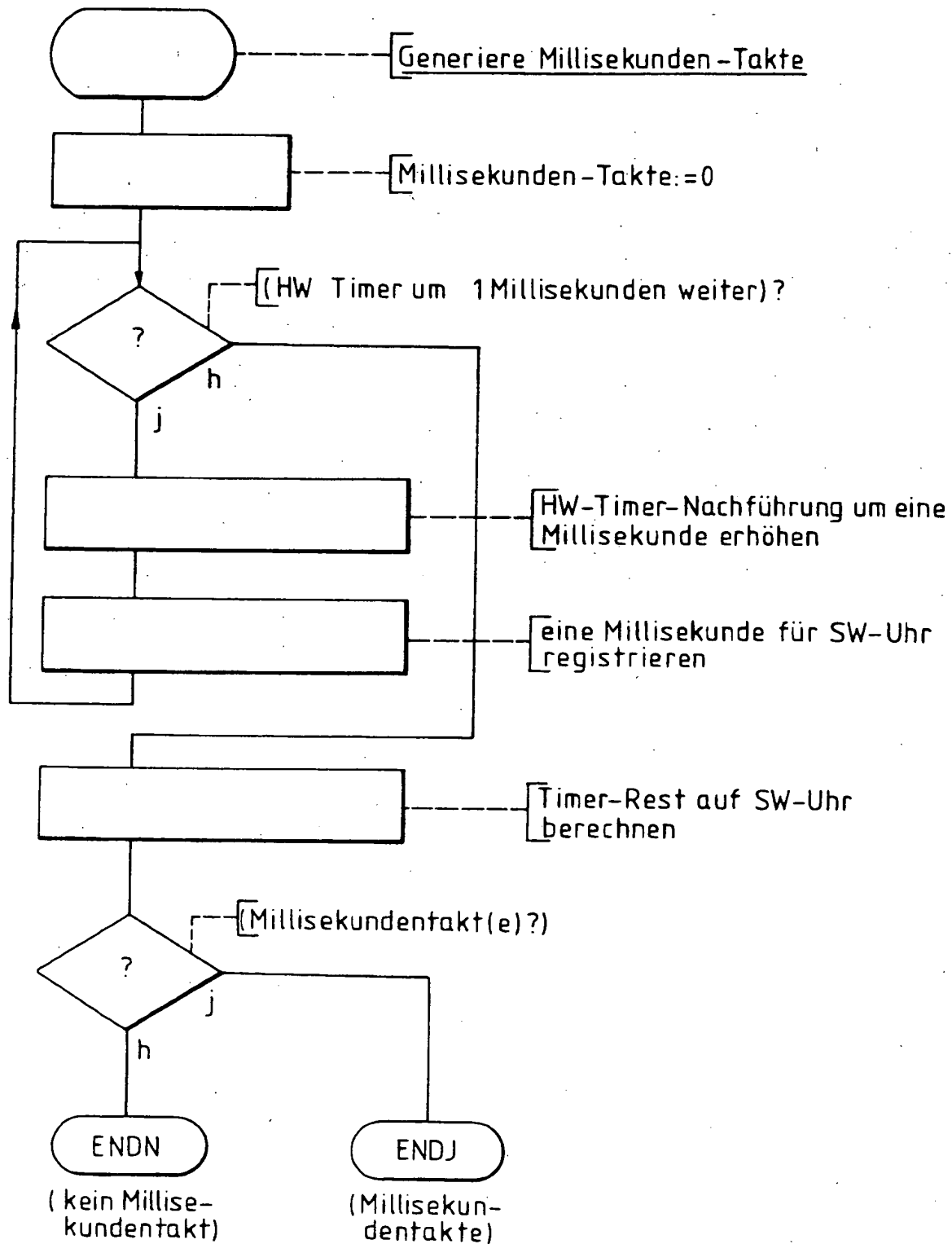


FIG.7



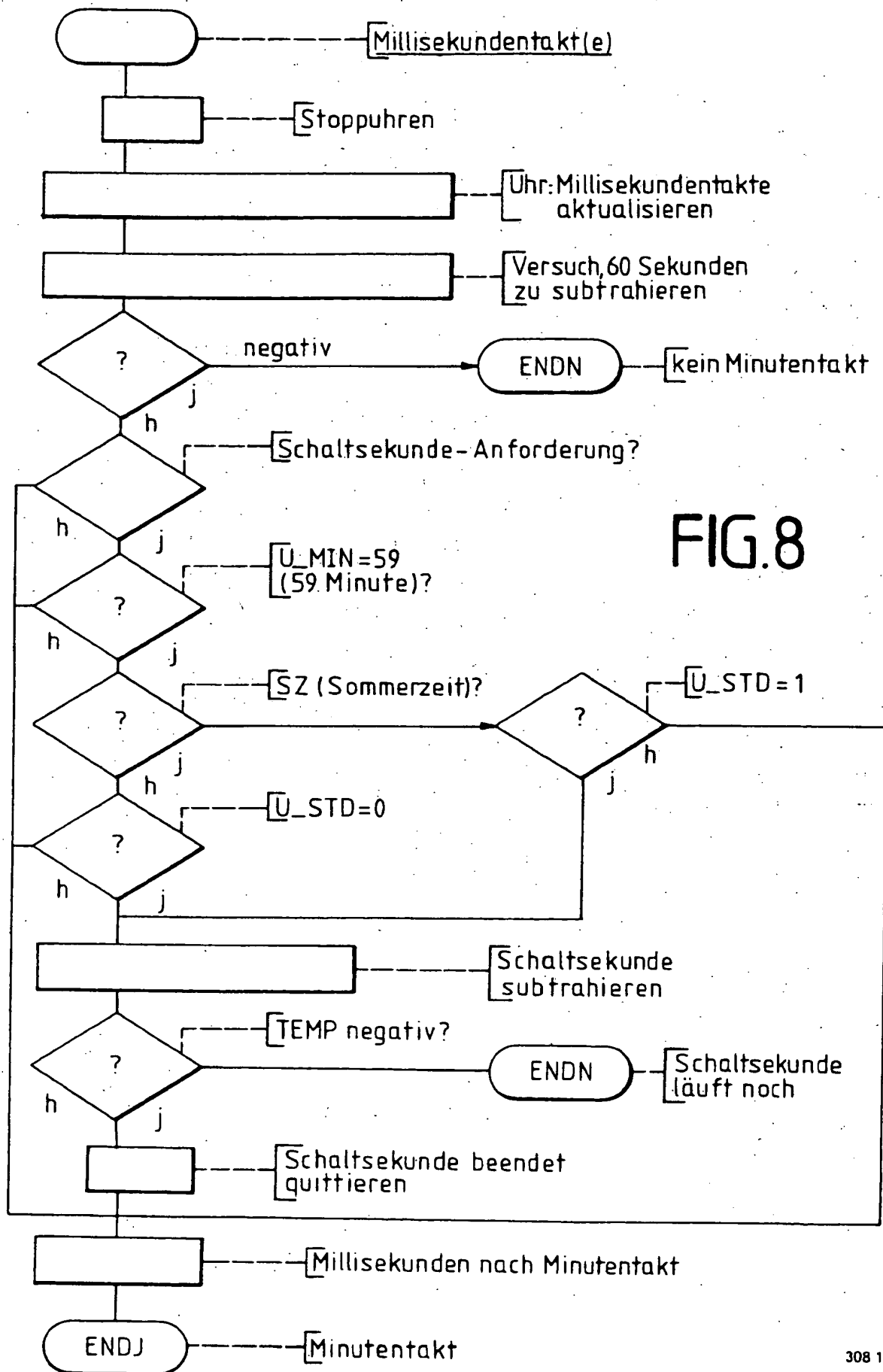
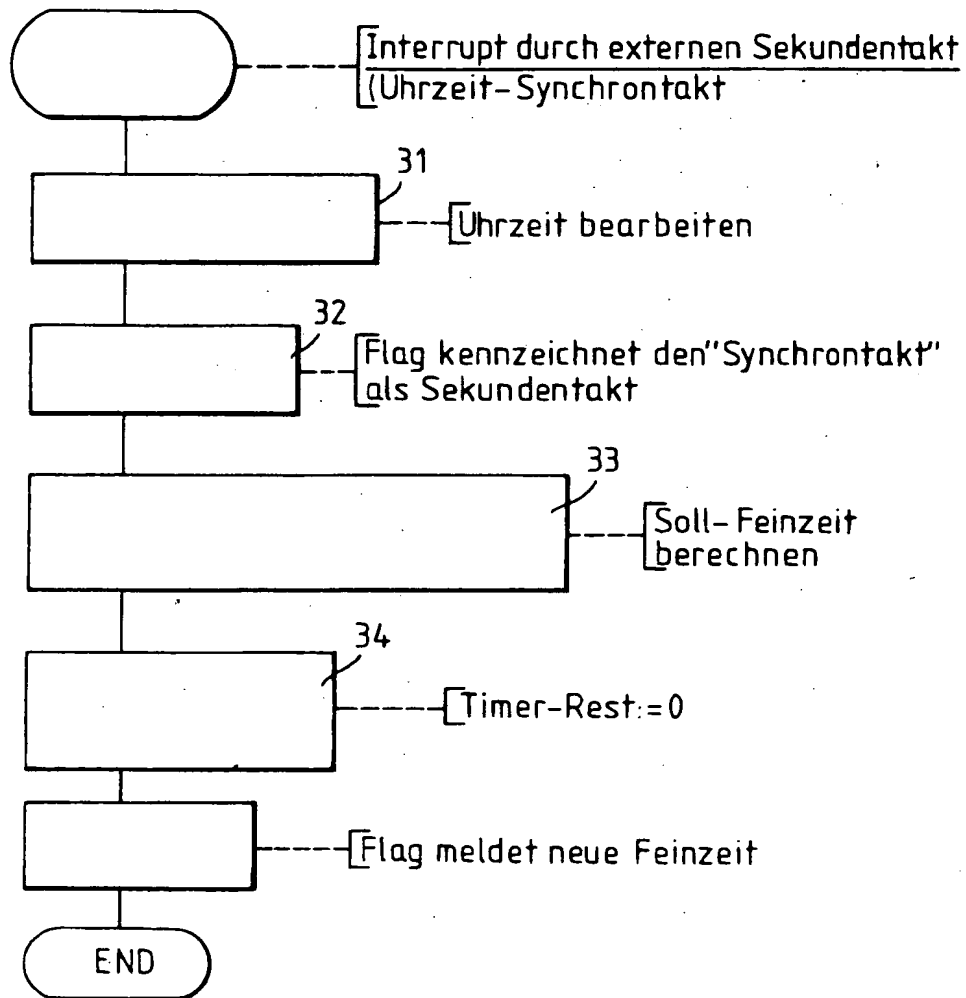


FIG. 9



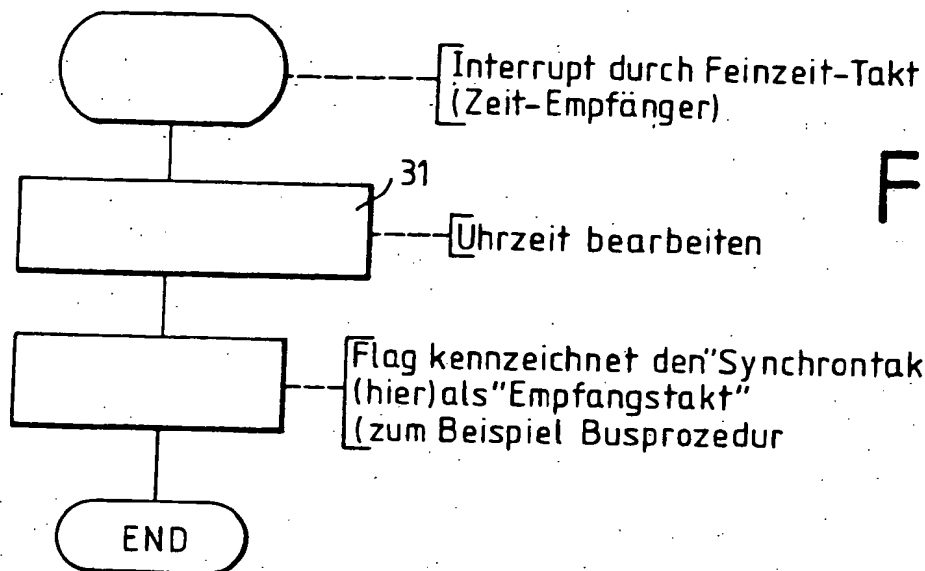


FIG.10

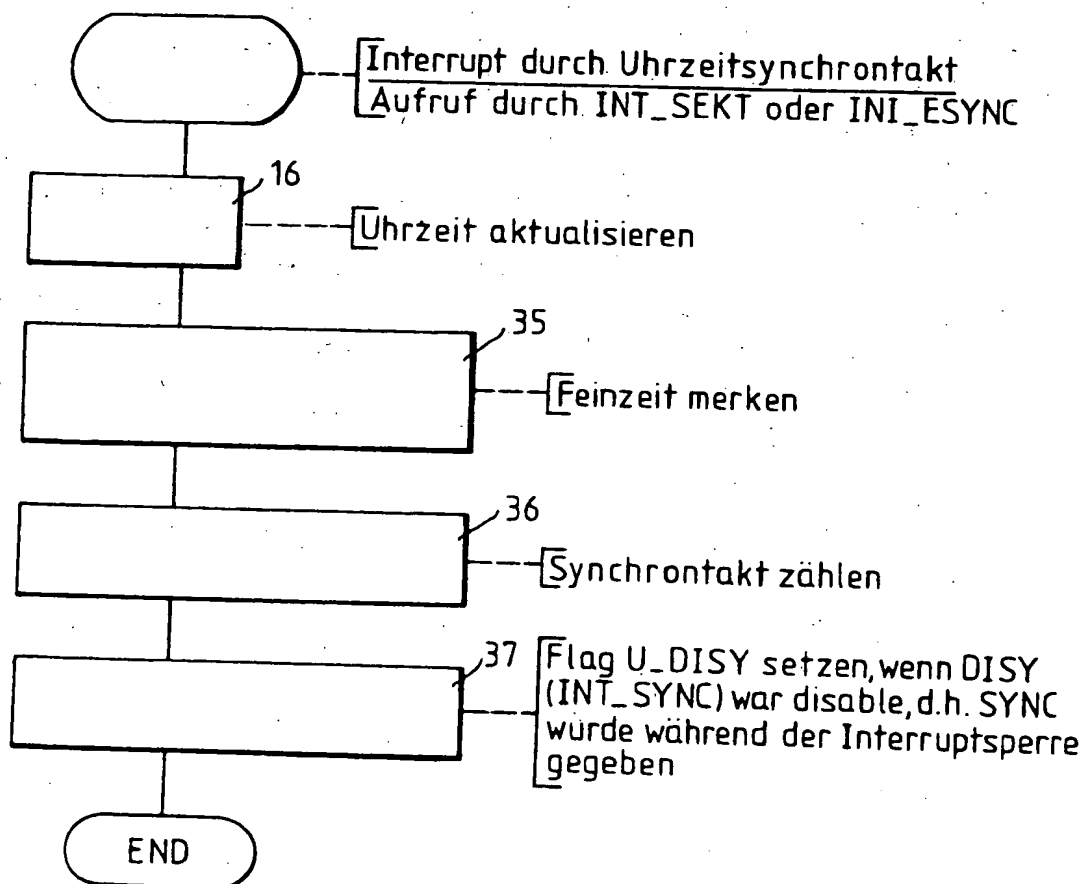
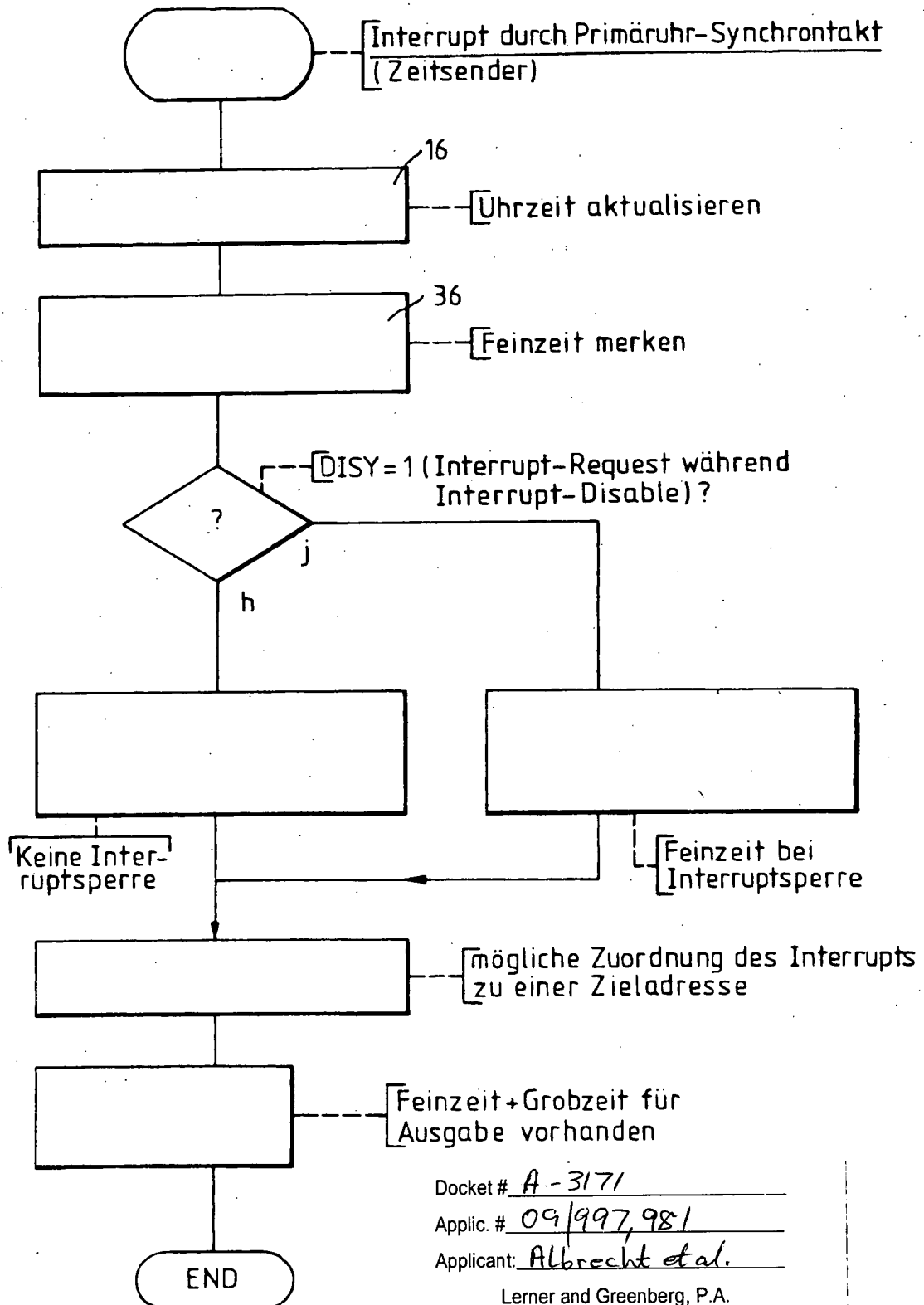


FIG.11

FIG.12

Docket # A-3171Applic. # 09/997,981Applicant: Albrecht et al.

Lerner and Greenberg, P.A.

Post Office Box 2480

Hollywood, FL 33022-2480

Tel: (954) 925-1100 Fax: (954) 925-1101

308 148/131